№ 2206

С.Ю. Муратова

Макросы и приложения

Курс лекций

№ 2206 МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»

Кафедра автоматизированных систем управления

С.Ю. Муратова

Макросы и приложения

Курс лекций

Рекомендовано редакционно-издательским
советом университета

МИСиС
ИЗДАТЕЛЬСКИЙ ДОМ

Москва 2012

УДК 681.3
М91

Р е ц е н з е н т
канд. техн. наук, доц. *А.И. Широков*

Муратова, С.Ю.
М91 Макросы и приложения : курс лекций / С.Ю. Муратова. – М. : Изд. Дом МИСиС, 2012. – 156 с.
ISBN 978-5-87623-623-4

Настоящий курс лекций разработан для семестрового модуля «Макросы и приложения» и посвящен разработке приложений на языке визуального программирования Visual Basic for Application в среде MS Excel.

Предназначен для бакалавров направлений 220700, 230100, 230400, 230700 и 231300.

УДК 681.3

ISBN 978-5-87623-623-4

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Одним из эффективных средств создания информационных систем или автоматизированных рабочих мест является программа MicroSoft Excel, которая предоставляет разработчику возможность использовать одновременно преимущества визуального программирования и электронной таблицы.

Языком визуального программирования в MicroSoft Excel, равно как в других приложениях MicroSoft Office, является Visual Basic for Applications (VBA). VBA можно отнести к языкам объектно-ориентированного программирования (ООП), в которых данные и код объединяются в нечто единое целое, называемое объектом.

Использование VBA актуально при разработке специфических приложений в малобюджетных организациях независимо от рода их деятельности, особенно если циркулирующая в них информация хранится в виде таблиц или баз данных Excel.

1. ОБЪЕКТЫ MS EXCEL

Основным понятием в MS Excel является объект. Говоря коротко, ***объект – это нечто, чем можно управлять и что можно программировать.*** Модель объектов Excel содержит более 100 собственных элементов и несколько – общих для всех приложений Office. Диапазон объектов Excel очень широк – от простых прямоугольников или текстовых полей до таких сложных структур, как сводные таблицы и диаграммы.

Каждый из объектов Excel предназначен для выполнения определенного действия, необходимого для анализа данных. Создание приложения заключается в объединении нужных объектов средствами языка программирования VBA.

1.1. Объекты, их свойства и методы

Каждый объект Excel располагает набором ***свойств*** (properties) и ***методов*** (methods).

Можно сказать, что свойства – это прилагательные, описывающие объект, а методы – глаголы, означающие действия, которые могут быть выполнены самим объектом или над ним.

1.1.1. Свойства: присвоение и использование значений

1.1.1.1. Присвоение значений

Рассмотрим в качестве примера объекта рабочую книгу (**Workbook**) – документ Excel. Вот некоторые свойства этого объекта[1]:

Свойство	Описание
Author	Имя пользователя, создавшего рабочую книгу
HasPassword	**True**, если рабочая книга защищена паролем, и **False** – в противном случае
Name	Название рабочей книги
Path	Путь к файлу книги на диске
Readonly	**True**, если сохранение рабочей книги запрещено, и **False** – в противном случае

[1] Полный перечень свойств можно просмотреть с помощью Object Browser.

Над свойством можно выполнять две операции: задать его значение или использовать его. И в том, и в другом случае необходимо указать имя объекта и имя свойства, разделив их точкой. Для определения значения свойства используется знак равенства (=). Например, инструкция для присваивания значения свойству **Author** рабочей книги **Мои таблицы.xls** выглядит так:

```
Workbooks("Мои таблицы.xls").Author = "Муратова С.Ю."
```

Структура этой строки такова:

`Workbooks("Мои таблицы.xls")`	Имя объекта;
`.`	Точка (разделитель);
`Author`	Имя свойства;
`=`	Знак присваивания;
`"Муратова С.Ю."`	Значение свойства.

ВНИМАНИЕ! Чтобы данная инструкция превратилась в подпрограмму VBA, перед ней нужно вставить строку со словом **Sub** и названием подпрограммы, а после нее – строку со словами **End Sub**, как показано ниже:

```
Sub ЗадатьАвтора()
    Dim Автор As String
    Workbooks("Мои таблицы.xls").Author _
    = "Муратова С.Ю."
End Sub
```

По мере знакомства со свойствами рабочей книги мы узнаем и несколько важных правил.

Во-первых, с каждым свойством связано значение, которое должно согласовываться с его типом (например, имя рабочей книги – строковое, свойство **HasPassword** – логическое и т.д.).

Во-вторых, свойство может принадлежать одному или нескольким различным объектам. Например, в Excel свойство **HasPassword** есть только у объекта **Workbook**, а вот свойство **Name** – практически у всех объектов Excel.

В-третьих, есть свойства, которые можно только использовать[1], но нельзя изменять. Таковым, например, является свойство **Path.**

[1]Так называемые свойства только для чтения (read-only).

1.1.1.2. Использование значений

Для использования значения свойства применяют ту же инструкцию VBA, что и для присвоения, но ее элементы располагают в обратном порядке. Чтобы «извлечь» значение свойства, его обычно присваивают какой-либо переменной. В строке программы, приведенной ниже, значение свойства **Author** объекта **Workbook** присваивается строковой переменной `Автор`.

```
Sub УзнатьАвтора()
    Dim Автор As String
    Автор = Workbooks("Мои таблицы.xls").Author
    MsgBox "Автор этой книги: " & Автор
End Sub
```

В данном фрагменте программы:

1) оператор **Dim** объявляет переменную `Автор` как строковую переменную;

2) оператор конкатенации **&** объединяет содержимое переменной `Автор` со строкой `"Автор этой книги:"` и отправляет полученное сообщение во встроенную функцию **MsgBox**, которая выводит его на экран.

1.1.2. Примеры использования методов рабочей книги Excel

Объект **Workbook** также содержит множество методов, определяющих действия, совершаемые им или над ним. Ниже приведены некоторые из них.

Метод	Действие
Activate	Активизация первого окна, связанного с книгой
Close	Закрытие книги
PrintPreview	Предварительный просмотр книги перед печатью
Protect	Защита книги паролем
Save	Сохранение книги

1.1.2.1. Вызов метода

Синтаксис команды VBA для вызова метода отличается от синтаксиса команды присвоения значения свойству. Все, что нужно в этом случае, – это указать объект и метод. Кроме того, в

большинстве методов Excel используются ***аргументы***, или ***параметры,*** – дополнительные данные для управления способом выполнения метода. Некоторые из аргументов необязательные, т.е. в зависимости от потребностей вы можете указывать все, некоторые или ни одного из них. Например, метод **Close** объекта **Workbook** имеет три необязательных аргумента:

Свойство	Значение
saveChanges	Принимает значения **True** (сохранить изменения в файле) или **False** (не сохранять изменения)
fileName	Имя файла для сохранения книги, если предыдущий аргумент имеет значение **True**
routeWorkbook	Принимает значения **True** (отправить книгу по маршруту) или **False** (не отправлять книгу)

При вызове метода без аргументов им присваиваются значения, заданные по умолчанию. Например, при отсутствии первого аргумента **saveChanges** в методе **Close** ему будет присвоено значение **True**. Аргумент **fileName** по умолчанию содержит текущее имя файла, а аргумент **routeWorkbook** – значение **False**. Вызов метода без аргументов запишется следующим образом:

```
Workbooks("Мои таблицы.xls").Close
```

1.1.2.2. Передача аргумента в метод

Существуют два способа передачи аргументов в метод: по позиции и по имени.

При передаче аргументов ***по позиции*** вы просто добавляете их к вызову метода, разделяя запятыми. ОБРАТИТЕ ВНИМАНИЕ на порядок следования аргументов! Для метода **Close**, например, правильный порядок таков: **saveChanges**, **fileName**, **routeWorkbook**.

Ниже приведен пример вызова метода **Close** со всеми тремя аргументами, переданными по позиции:

```
Workbooks("Мои таблицы.xls").Close True, _
          "Мои таблицы-2.xls",  False
```

В данном примере метод **Close** закрывает рабочую книгу `Мои таблицы.xls`, сохраняя её под именем `Мои таблицы-2.xls`.

ОБРАТИТЕ ВНИМАНИЕ на символ подчеркивания в конце первой строки. Он означает, что команда продолжается на следующей строке. Перед символом подчеркивания необходимо ввести пробел!

Чтобы пропустить какой-то аргумент (например, второй), вставьте вместо него пробел:

```
Workbooks("Мои таблицы.xls").Close True, , False
```

При передаче аргумента ***по имени*** надо указать в вызове три элемента:

- имя аргумента (например, **saveChanges**);
- оператор присваивания с двоеточием (**:=**);
- значение аргумента.

В следующем примере метод **Close** вызывается с аргументами**,** передаваемыми по имени:

```
Workbooks("Мои таблицы.xls").Close _
saveChanges:=True, fileName:="Мои таблицы-2.xls", _
routeWorkbook:=False
```

Передавая аргументы по имени, не обязательно соблюдать их порядок. Приведенный ниже код идентичен предыдущему:

```
Workbooks("Мои таблицы.xls").Close _
routeWorkbook:=False, savChanges:=True, _
fileName:="Мои таблицы-2.xls"
```

При передаче аргументов по имени программа становится более понятной, а по позиции – более короткой.

1.2. Ссылки на одиночные объекты и объекты из семейств

Как уже говорилось, при обращении к свойству или методу объекта их имена просто добавляют к имени объекта, отделяя от него точкой. Ссылка же на сам объект может выглядеть по-разному. К одиночному объекту ***семейства*** обращаются либо по имени, либо по его номеру в ***семействе*** объектов. Чтобы разобраться, чем отличаются два этих способа, уясним себе, что такое ***семейство*** (collection). Если не вдаваться в детали, ***семейство – это группа похожих объектов.*** Все объекты Excel можно отнести к одному из двух приблизительно равных по численности классов: ***одиночным***

объектам и объектам из семейства[1]. К первым обращаются непосредственно, ко вторым – по имени или по номеру. В предыдущих примерах обращение к рабочей книге `Мои таблицы.xls` было оформлено с помощью указателя на объект семейства **Workbooks**[2] по имени.

1.2.1. Различия между одиночными объектами и объектами из семейств

Как же узнать, является объект одиночным или входит в семейство? Для этого надо применить простые, интуитивно понятные правила.

Правило 1. Одиночный объект может существовать в данный момент времени только в одном экземпляре. Другими словами, он уникален.

Правило 2. Объекты из семейства могут существовать одновременно в нескольких экземплярах.

В Excel может быть открыто несколько рабочих книг, поэтому к объекту `Мои таблицы.xls` можно было бы, согласно правилу 2, обратиться по индексу, предварительно узнав его порядковый номер. Следующий пример демонстрирует эту возможность:

```
Sub ССылкаНаКнигуПоИндексу()
'1. Объявляем переменные:
    Dim Автор              As String
    Dim КолОткрытыхКниг    As Integer
'2. Определяем количество открытых книг:
    КолОткрытыхКниг = Workbooks.Count
'3. Ищем среди открытых книг книгу с именем Мои
'таблицы.xls:
For i = 1 To КолОткрытыхКниг
   If Workbooks(i).Name = "Мои таблицы.xls" Then
      Автор = Workbooks(i).Author
```

---

[1] Из этой классификации слегка выбиваются три объекта Excel: Range, Sheets и Shapes. Объект Range считается одиночным, но обладает также характеристиками семейства. Sheets и Shapes — семейства, но не объектов, а других семейств.

[2] Семейство рабочих книг

```
        MsgBox "Автор этой книги: " & Автор
        Exit Sub
    End If
Next i
MsgBox "Книга Мои таблицы.xls среди открытых книг не найдена!"
End Sub
```

В этом примере обратите внимание на комментарии – текст, расположенный после апострофа ('). Программу с пояснительными комментариями значительно легче отлаживать и вносить в неё изменения. Помните, что в программе без комментариев её детали забываются через удивительно короткое время!

ЗАМЕЧАНИЯ. 1. Комментарии должны содержать некоторую дополнительную информацию, а не перефразировать программу. С этой точки зрения 1-й и 2-й комментарии правомерны как пояснения только в учебной программе.

2. Обязательно делайте вводные комментарии. Они должны включать в себя следующие пункты:

- назначение программы;
- указания по вызову программы и её использованию;
- список и назначение основных переменных и массивов;
- указания по вводу-выводу, список всех файлов;
- список используемых подпрограмм и их назначение;
- требования к компьютеру;
- сведения об авторе;
- дату написания программы.

3. Для улучшения наглядности программы вставляйте пустые строки и делайте отступы.

1.2.2. Семейства как объекты

Хотя семейство объединяет несколько объектов, оно и само является объектом, причем одиночным. Концепцию семейства лучше проиллюстрировать на конкретном примере.

Семейство **Workbooks** содержит несколько объектов **Workbook**, с каждым из которых связан набор свойств и методов. Другой набор свойств и методов связан с семейством **Workbooks**. Таким образом, в Excel есть три вида объектов: ***одиночный объект, объект семей-***